Παράρτημα – Ολοκληρωμένη Εφαρμογή

Εφαρμογή On-line Παραγγελίες

Περιγραφή

Η εφαρμογή που παρατίθεται είναι μία εφαρμογή Business to Business (B2B) στην οποία μία εταιρεία που εμπορεύεται διάφορα προϊόντα μεταπουλώντας τα σε άλλες εταιρείες (χονδρικής), θέλει να παρέχει μέσω του Ιστοτόπου της την δυνατότητα σε κάποιο πελάτη να δει τα προϊόντα και να μπορεί να εισάγει παραγγελίες.
Επίσης να μπορεί υπάλληλος της εταιρείας να ενημερώνει τα προϊόντα, όσον αφορά τη διαθέσιμη ποσότητα. Στόχος είναι οι σελίδες αυτές να είναι προστατευμένες και να επιτρέπεται η χρήση τους μόνο μετά από την πιστοποίηση του χρήστη.

Προδιαγραφές

Να αναπτυχθεί εφαρμογή για on-line διαχείριση παραγγελιών χρησιμοποιώντας PHP – MySQL. Η εφαρμογή on-line παραγγελιών θα επιτρέπει την πρόσβαση σε διαχειριστές και πελάτες που έχουν προηγούμενα πιστοποιηθεί (κάνοντας login). Η πιστοποίηση εξουσιοδοτημένων χρηστών θα πρέπει να γίνεται με τέτοιο τρόπο ώστε να προστατεύονται όχι μόνο μία αλλά όλες οι σελίδες με "προστατευμένο" περιεχόμενο. Η προστασία να γίνει κάνοντας χρήση συνόδων (sessions).
Θα πρέπει οι Διαχειριστές να μπορούν να προσθέτουν νέα προϊόντα καθώς και να προσθέτουν ποσότητα σε υπάρχοντα προϊόντα.
Οι πελάτες θα πρέπει να μπορούν να εγγράφονται ώστε να αποκτούν πρόσβαση (δηλαδή username και password), και να εκτελούν παραγγελίες επιλέγοντας ένα υπάρχον προϊόν και συμπληρώνοντας μια ποσότητα.
Εκτός των PHP scripts, θα χρειαστεί να σχεδιάσετε τις ακόλουθες φόρμες (HTML):
Φόρμες για διαχειριστές:

1. Εισαγωγή νέου προϊόντος (όνομα προϊόντος, αρχικό stock)
2. Προσθήκη κάποιων μονάδων στο διαθέσιμο stock ενός προϊόντος.

Η πρόσβαση και στις δύο παραπάνω φόρμες είναι διαθέσιμη μετά από πιστοποίηση του διαχειριστή.
Φόρμες για πελάτες:

1. Εισαγωγή στοιχείων πελάτη (επώνυμο, όνομα, διεύθυνση, username, password). Η πρόσβαση σε αυτή τη φόρμα είναι διαθέσιμη χωρίς πιστοποίηση.
2. Προσθήκη μιας παραγγελίας (πελάτης στον οποίο καταχωρείται η παραγγελία, προϊόν που παραγγέλνεται, ποσότητα παραγγελίας σε μονάδες). Η πρόσβαση σε αυτή τη φόρμα είναι διαθέσιμη μετά από πιστοποίηση του πελάτη. Φυσικά, μετά την καταχώριση της παραγγελίας, το stock του προϊόντος που παραγγέλθηκε μειώνεται κατά τόσες μονάδες όσες η ποσότητα που ζητήθηκε. Θα πρέπει επίσης να προηγηθεί της καταχώρισης της παραγγελίας ένας έλεγχος ότι η ποσότητα που ζητείται είναι διαθέσιμη.

Σχεδίαση-Υλοποίηση Βάσης Δεδομένων

Δημιουργία Βάσης Δεδομένων με phpMyAdmin

Για την εφαρμογή σχεδιάστηκε και δημιουργήθηκε μια Βάση Δεδομένων σε MySQL Server μέσω του phpMyAdmin. Το όνομα της βάσης είναι ordersdb και αποτελείται από τους επόμενους τέσσερις πίνακες:

1. **administrators** (πίνακας που αποθηκεύονται τα στοιχεία των διαχειριστών)
2. **customers** (πίνακας που αποθηκεύονται τα στοιχεία των πελατών)
3. **products** (πίνακας που αποθηκεύονται τα στοιχεία των προϊόντων)
4. **orders** (πίνακας που αποθηκεύονται τα στοιχεία των παραγγελιών των πελατών)

Στο επόμενο σχήμα φαίνεται και η δομή της ΒΔ από το phpMyAdmin.

Πίνακας	Ενέργεια	Εγγραφές	Τύπος	Μέγεθος
administrators		1	MyISAM	2,0 KB
customers		0	MyISAM	1,0 KB
orders		0	MyISAM	1,0 KB
products		0	MyISAM	1,0 KB

Πίνακας **administrators**

Στον πίνακα αυτόν αποθηκεύουμε το όνομα χρήστη και τον κωδικό των διαχειριστών. Ο πίνακας αυτός έχει δύο πεδία:

Πεδίο	Τύπος	Χαρακτηριστικά	Κλειδί	Χρήση
username	Varchar(20)	Not null	Yes	Αποθηκεύουμε το όνομα χρήστη του διαχειριστή.
password	Varchar(20)	Not null	No	Αποθηκεύουμε τον κωδικό του διαχειριστή

Στο επόμενο σχήμα φαίνεται και η δομή του πίνακα από το phpMyAdmin.

Πεδίο	Τύπος	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
username	varchar(20)		Όχι			
password	varchar(20)		Όχι			

Στον πίνακα αυτόν έχει ήδη εισαχθεί το ζεύγος τιμών (‘admin1’, ‘pass1’) ώστε να μπορεί να μπει ένας διαχειριστής στην εφαρμογή και να προσθέσει προϊόντα.

Πίνακας **customers**

Στον πίνακα αυτόν αποθηκεύουμε τα στοιχεία των χρηστών - πελατών. Ο πίνακας αυτός έχει έξι πεδία:

Πεδίο	Τύπος	Χαρακτηριστικά	Κλειδί	Χρήση
customer_id	int	Not null, auto_increment	Yes	Αυτόματη τιμή που χρησιμοποιείται για να καθορίσει μοναδικά τον πελάτη. Αποτελεί και ξένο κλειδί στον πίνακα orders.
lastname	Varchar(30)	Not null	No	Αποθηκεύουμε το επώνυμο του πελάτη.
firstname	Varchar(30)	Not null	No	Αποθηκεύουμε το όνομα του πελάτη.
address	Varchar(50)	Not null	No	Αποθηκεύουμε τη διεύθυνση του πελάτη.
username	Varchar(20)	Not null	No	Αποθηκεύουμε το όνομα χρήστη του πελάτη.
password	Varchar(20)	Not null	No	Αποθηκεύουμε τον κωδικό του πελάτη

Παρόλο που το όνομα χρήστη δεν είναι κλειδί, στην χρήση αυτού του πίνακα κατά την εισαγωγή δεν επιτρέπω να μπει δεύτερο ίδιο όνομα χρήστη. Στο επόμενο σχήμα φαίνεται και η δομή του πίνακα από το phpMyAdmin.

Πεδίο	Τύπος	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
customer_id	int(11)		Όχι		auto_increment	
lastname	varchar(30)		Όχι			
firstname	varchar(30)		Όχι			
address	varchar(50)		Όχι			
username	varchar(20)		Όχι			
password	varchar(20)		Όχι			

Στον πίνακα αυτόν έχει ήδη εισαχθεί ένας πελάτης μέσω της φόρμας εγγραφής πελάτη της εφαρμογής με όνομα χρήστη **user1** και κωδικό **pass1**.

Πίνακας **products**

Στον πίνακα αυτόν αποθηκεύουμε τα στοιχεία των προϊόντων. Ο πίνακας αυτός έχει τρία πεδία:

Πεδίο	Τύπος	Χαρακτηριστικά	Κλειδί	Χρήση
product_id	int	Not null, auto_increment	Yes	Αυτόματη τιμή που χρησιμοποιείται για να καθορίσει μοναδικά το προϊόν. Αποτελεί και ξένο κλειδί στον πίνακα orders.
product	Varchar(30)	Not null	No	Αποθηκεύουμε την περιγραφή του προϊόντος.
stock	Int	Not null	No	Αποθηκεύουμε την ποσότητα του προϊόντος που είναι διαθέσιμη για παραγγελία.

Παρόλο που την περιγραφή του δεν την έχω κλειδί, στην χρήση αυτού του πίνακα κατά την εισαγωγή δεν επιτρέπω να μπει δεύτερο ίδιο προϊόν. Στο επόμενο σχήμα φαίνεται και η δομή του πίνακα από το phpMyAdmin.

Πεδίο	Τύπος	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
product_id	int(11)		Όχι		auto_increment	
product	varchar(50)		Όχι			
stock	int(11)		Όχι	0		

Στον πίνακα αυτόν έχουν ήδη εισαχθεί δύο προϊόντα μέσω της φόρμας εισαγωγής προϊόντων του Διαχειριστή. Τα προϊόντα είναι **HP DeskJet 3650** και **HP LaserJet 1100**.

Πίνακας **orders**

Στον πίνακα αυτόν αποθηκεύουμε τα στοιχεία των παραγγελιών. Ο πίνακας αυτός έχει τέσσερα πεδία:

Πεδίο	Τύπος	Χαρακτηριστικά	Κλειδί	Χρήση
order_id	int	Not null, auto_increment	Yes	Αυτόματη τιμή που χρησιμοποιείται για να καθορίσει μοναδικά την παραγγελία.
customer_id	int	Not null	No	Είναι ξένο κλειδί από τον πίνακα customers και καθορίζει ποιος πελάτης έκανε την παραγγελία.
product_id	int	Not null	No	Είναι ξένο κλειδί από τον πίνακα products και καθορίζει ποιο προϊόν

				παραγγέλθηκε.
quantity	int	Not null	No	Η ποσότητα παραγγελίας του προϊόντος.

Πεδίο	Τύπος	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
order_id	int(11)		Όχι		auto_increment	
customer_id	int(11)		Όχι	0		
product_id	int(11)		Όχι	0		
quantity	int(11)		Όχι	0		

Στον πίνακα αυτόν έχουν ήδη εισαχθεί μερικές παραγγελίες από τη φόρμα εισαγωγής παραγγελιών του πελάτη..

Εντολές SQL δημιουργίας Βάσης Δεδομένων

Ακολουθούν οι εντολές SQL με τις οποίες μπορούμε να κατασκευάσουμε την παραπάνω Βάση Δεδομένων.

```
# Βάση : ordersdb
CREATE Database ordersdb;

# Δομή Πίνακα για τον Πίνακα `administrators`
CREATE TABLE `administrators` (
  `username` varchar(20) NOT NULL default '',
  `password` varchar(20) NOT NULL default '',
  PRIMARY KEY  (`username`)
) TYPE=MyISAM;


# Εισαγωγή στοιχείων Διαχειριστή
INSERT INTO `administrators` VALUES ('admin1', 'pass1');


# Δομή Πίνακα για τον Πίνακα `customers`
CREATE TABLE `customers` (
  `customer_id` int(11) NOT NULL auto_increment,
  `lastname` varchar(30) NOT NULL default '',
  `firstname` varchar(30) NOT NULL default '',
  `address` varchar(50) NOT NULL default '',
  `username` varchar(20) NOT NULL default '',
  `password` varchar(20) NOT NULL default '',
  PRIMARY KEY  (`customer_id`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;


# Δομή Πίνακα για τον Πίνακα `orders`
CREATE TABLE `orders` (
  `order_id` int(11) NOT NULL auto_increment,
  `customer_id` int(11) NOT NULL default '0',
```

```
  `product_id` int(11) NOT NULL default '0',
  `quantity` int(11) NOT NULL default '0',
  PRIMARY KEY  (`order_id`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;


# Δομή Πίνακα για τον Πίνακα `products`
CREATE TABLE `products` (
  `product_id` int(11) NOT NULL auto_increment,
  `product` varchar(50) NOT NULL default '',
  `stock` int(11) NOT NULL default '0',
  PRIMARY KEY  (`product_id`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;
```

Σχεδίαση-Υλοποίηση Ιστοσελίδων-Κώδικα

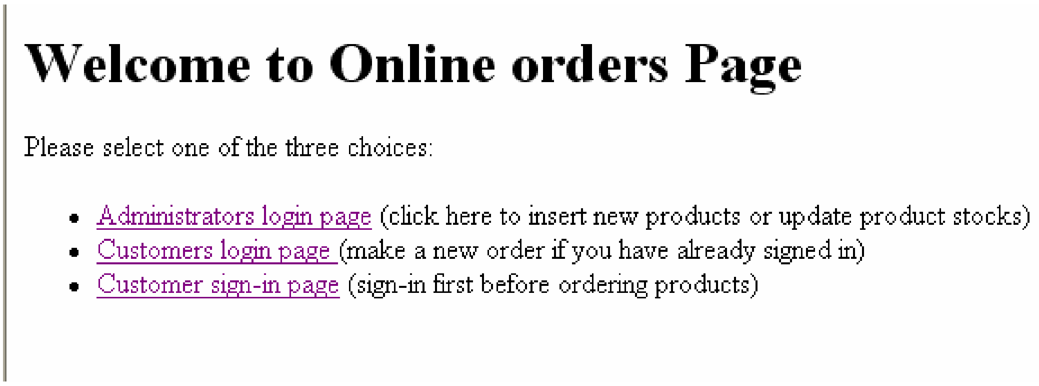
Για τις απαιτήσεις της εργασίας δημιουργήθηκαν οι επόμενες ιστοσελίδες/αρχεία.

functions.php

Για ευκολία και επαναχρησιμοποίηση του κώδικα όλες οι συναρτήσεις php που αφορούν την πιστοποίηση πελατών και διαχειριστών και όλες τις άλλες λειτουργίες της εφαρμογής έχουν γραφτεί στο αρχείο **functions.php** το οποίο χρησιμοποιούν οι ιστοσελίδες για καλέσουν τις συναρτήσεις που χρειάζονται. Στο αρχείο αυτό υπάρχουν σχόλια για το τι κάνει κάθε συνάρτηση.

index.htm

Η ιστοσελίδα αυτή παρουσιάζει την αρχική και κυρίως ιστοσελίδα της εφαρμογής. Ο χρήστης μπορεί να επιλέξει από τρεις επιλογές όπως φαίνεται και στο επόμενο σχήμα:



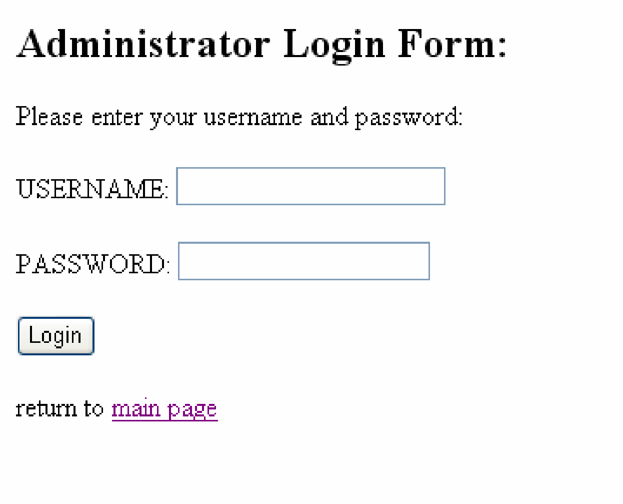
<u>**Administrators login page**</u>: Από την πρώτη επιλογή ο χρήστης μπορεί να μπει στο σύστημα αφού εισάγει κατάλληλο όνομα χρήστη και κωδικό σαν administrator ώστε να μπορέσει να εισάγει νέα προϊόντα ή να ενημερώσει το stock σε υπάρχοντα προϊόντα (αυξάνοντάς το με μια ποσότητα). Η πρώτη αυτή επιλογή σε οδηγεί στην ιστοσελίδα **adminlogin.php**

<u>**Customers login page**</u>: Από την δεύτερη επιλογή ο χρήστης μπορεί να μπει στο σύστημα αφού εισάγει κατάλληλο όνομα χρήστη και κωδικό σαν πελάτης ώστε να μπορέσει να κάνει παραγγελίες για υπάρχοντα προϊόντα (επιλέγοντας το προϊόν και δίνοντας μια ποσότητα). Η δεύτερη αυτή επιλογή σε οδηγεί στην ιστοσελίδα **custlogin.php**

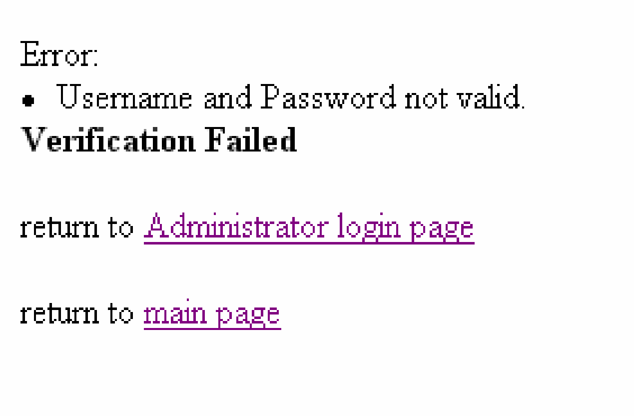
Customers sign-in page: Από την τρίτη επιλογή ο χρήστης μπορεί να κάνει εγγραφή στο σύστημα (δίνοντας τα στοιχεία του και ένα ζεύγος ονόματος χρήστη και κωδικού). Αφού κάνει εγγραφή μπορεί με την δεύτερη επιλογή να κάνει παραγγελίες χρησιμοποιώντας το όνομα χρήστη και τον κωδικό της εγγραφής του. Η τρίτη αυτή επιλογή σε οδηγεί στην ιστοσελίδα **custregform.htm**

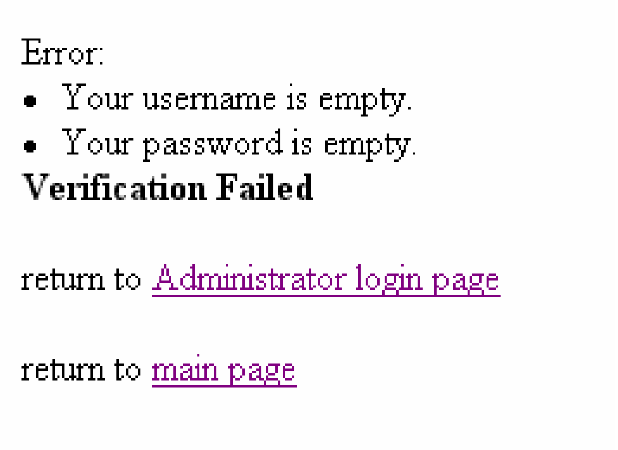
adminlogin.php

Η ιστοσελίδα αυτή παρουσιάζει την φόρμα εισαγωγής στο σύστημα ενός administrator. Η φόρμα έχει τη μορφή που φαίνεται στο επόμενο σχήμα:



Ο χρήστης εδώ πρέπει να δώσει ένα όνομα χρήστη και έναν κωδικό διαχειριστή για να μπορέσει να δει τις σελίδες των διαχειριστών. Η φόρμα αποστέλλεται στο αρχείο **adminverify.php**. Αν δοθεί κάτι λάθος το αρχείο αυτό σου βγάζει ένα λάθος και σου δίνει ένα σύνδεσμο για να επιστρέψεις στη φόρμα. Στα επόμενα δύο σχήματα έχουμε δύο περιπτώσεις όπου στην πρώτη δεν δόθηκε σωστό όνομα χρήστη-κωδικός και στη δεύτερη δεν συμπληρώθηκαν καθόλου τα πεδία. Σε όλες τις περιπτώσεις δίνεται πάντα και ένας σύνδεσμος για επιστροφή στην κυρίως σελίδα.





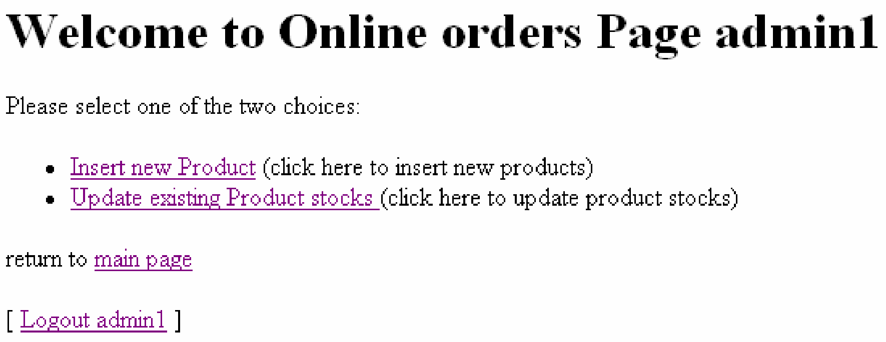
adminverify.php

Η ιστοσελίδα αυτή ελέγχει την είσοδο στο σύστημα ενός διαχειριστή χρησιμοποιώντας τη συνάρτηση **adminlogin_check()** από το αρχείο **functions.php**. Σε περίπτωση λάθους εμφανίζει ένα μήνυμα περιγραφής του λάθους και σου δίνει δύο συνδέσμους, έναν για να ξαναγυρίσεις στην φόρμα και να

ξαναπροσπαθήσεις και έναν για να επιστρέψεις στην αρχική σελίδα (όπως φαίνεται στα δύο προηγούμενα σχήματα). Αν όλα πάνε καλά εμφανίζει την φόρμα με τις επιλογές του διαχειριστή που είναι το αρχείο **adminindex.php**.

adminindex.php

Η ιστοσελίδα αυτή παρουσιάζει την αρχική σελίδα επιλογών ενός διαχειριστή όπως φαίνεται και στο επόμενο σχήμα. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής). Ο διαχειριστής μπορεί να επιλέξει από δύο επιλογές όπως φαίνεται και στο επόμενο σχήμα:

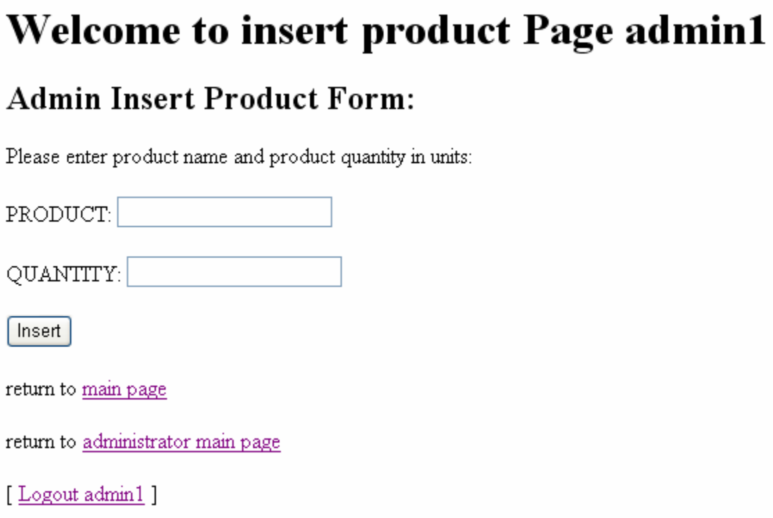


Insert new Product: Από την πρώτη επιλογή ο administrator εισάγει νέα προϊόντα. Η πρώτη αυτή επιλογή σε οδηγεί στην ιστοσελίδα **insertproducts.php**

Update existing Product stocks: Από την δεύτερη επιλογή ο administrator ενημερώνει το stock κάποιου προϊόντος που υπάρχει ήδη. Η δεύτερη αυτή επιλογή σε οδηγεί στην ιστοσελίδα **updateproducts.php**
Επίσης υπάρχουν και οι επιλογές για να βγει από διαχειριστής (logout) και για να επιστρέψει στην αρχική σελίδα της εφαρμογής.

insertproducts.php

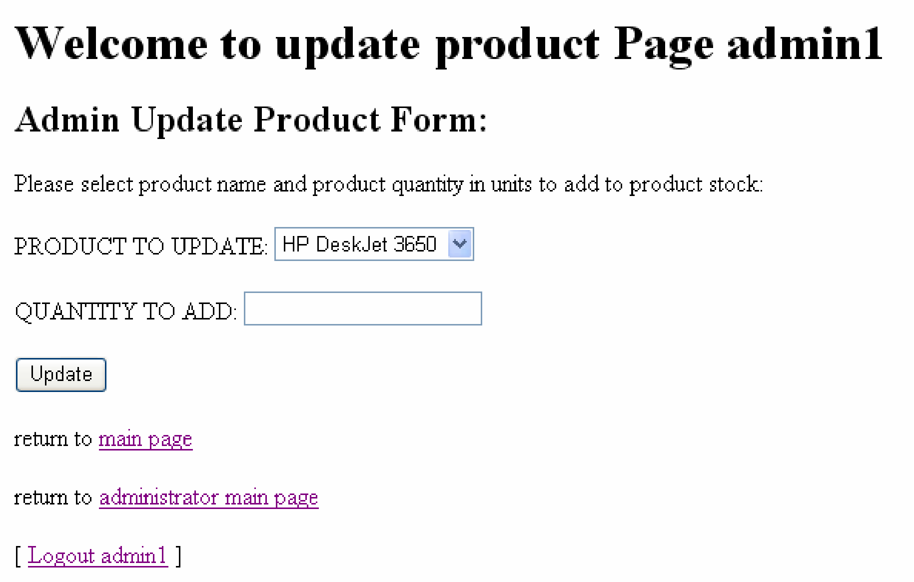
Η ιστοσελίδα αυτή παρουσιάζει μία φόρμα για να εισάγει ο διαχειριστής ένα νέο προϊόν όπως φαίνεται και στο επόμενο σχήμα. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής). Ο διαχειριστής πρέπει να εισάγει το όνομα του προϊόντος και την ποσότητα. Η φόρμα αποστέλλεται στο αρχείο **verifyproductinsertion.php**.



Επίσης υπάρχουν και οι επιλογές για να βγει από διαχειριστής (logout), να επιστρέψει στην σελίδα των επιλογών των διαχειριστών και για να επιστρέψει στην αρχική σελίδα της εφαρμογής.

updateproducts.php

Η ιστοσελίδα αυτή παρουσιάζει μία φόρμα για να ενημερώσει ο διαχειριστής ένα υπάρχον προϊόν όπως φαίνεται και στο επόμενο σχήμα. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής). Επίσης χρησιμοποιεί και την συνάρτηση **getproducts()** από το ίδιο αρχείο για να προβάλει σε μια λίστα όλα τα υπάρχοντα προϊόντα. Ο διαχειριστής πρέπει να επιλέξει το όνομα του προϊόντος και να εισάγει την ποσότητα. Η φόρμα αποστέλλεται στο αρχείο **verifyproductupdate.php**.



Επίσης υπάρχουν και οι επιλογές για να βγει σαν διαχειριστής (logout), να επιστρέψει στην σελίδα των επιλογών των διαχειριστών και για να επιστρέψει στην αρχική σελίδα της εφαρμογής.

verifyproductinsertion.php

Η ιστοσελίδα αυτή ελέγχει αν μπορεί να γίνει η εισαγωγή ενός νέου προϊόντος με κάποιο stock. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής). Επίσης χρησιμοποιεί και την συνάρτηση **insertproduct_check()** από το ίδιο αρχείο για να κάνει τον έλεγχο και την εισαγωγή αν μπορεί να γίνει. Στην περίπτωση που υπάρχει κάποιο λάθος προβάλει μια περιγραφή του και σου δίνει τις επιλογές να ξαναγυρίσεις στην φόρμα εισαγωγής προϊόντος, να γυρίσεις στην αρχική σελίδα και την επιλογή να βγεις από διαχειριστής. Δύο περιπτώσεις λάθους (δεν δόθηκαν στοιχεία και δόθηκε ήδη υπάρχον προϊόν) φαίνονται στα δύο επόμενα σχήματα.

- The product already exist, please use Product Update Form to update stock of existing products.

Product Insertion Failed

The insertion Failed

return to insert product page

return to main page

[Logout admin1]

Στην περίπτωση που η εισαγωγή πετύχει, προβάλλει ένα μήνυμα επιτυχίας και σου δίνει τις επιλογές να ξαναγυρίσεις στην φόρμα ενημέρωσης προϊόντος, να γυρίσεις στην αρχική σελίδα και την επιλογή να βγεις από διαχειριστής, όπως φαίνεται στο επόμενο σχήμα.

The insertion was succesfull

Product: HP DeskJet 7250
Quantity: 10

return to insert product page

return to main page

[Logout admin1]

verifyproductupdate.php

Η ιστοσελίδα αυτή ελέγχει αν μπορεί να γίνει η ενημέρωση του stock ενός υπάρχοντος προϊόντος αυξάνοντάς το με κάποιο ποσότητα. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής). Επίσης χρησιμοποιεί και την συνάρτηση **updateproduct_check()** από το ίδιο αρχείο για να κάνει τον έλεγχο και την εισαγωγή αν μπορεί να γίνει. Στην περίπτωση που υπάρχει κάποιο λάθος προβάλει μια περιγραφή του και σου δίνει τις επιλογές να ξαναγυρίσεις στην φόρμα ενημέρωσης προϊόντος, να γυρίσεις στην αρχική σελίδα και την επιλογή να βγεις από διαχειριστής. Δύο περιπτώσεις λάθους (δεν δόθηκε ποσότητα και δόθηκε ποσότητα 0) φαίνονται στα δύο επόμενα σχήματα.

- The quantity is empty.

Product Update Failed

The updateFailed

return to update product page

return to main page

[Logout admin1]

- The quantity must be greater than 0

Product Update Failed

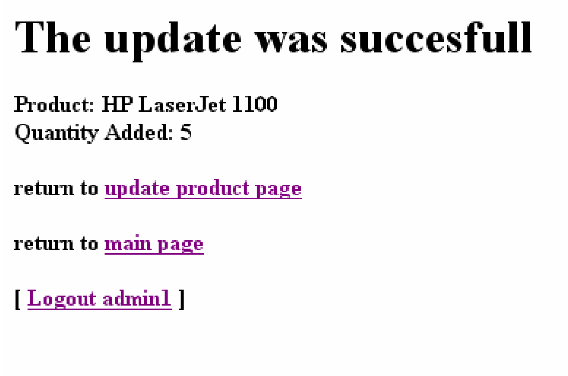
The updateFailed

return to update product page

return to main page

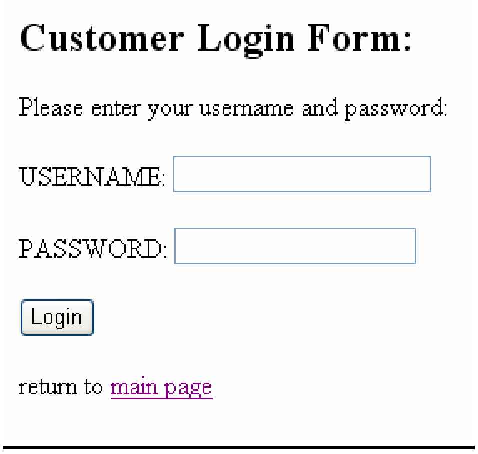
[Logout admin1]

Στην περίπτωση που η ενημέρωση πετύχει σου προβάλλει ένα μήνυμα επιτυχίας και σου δίνει τις επιλογές να ξαναγυρίσεις στην φόρμα ενημέρωσης προϊόντος, να γυρίσεις στην αρχική σελίδα και την επιλογή να βγεις από διαχειριστής, όπως φαίνεται στο επόμενο σχήμα.

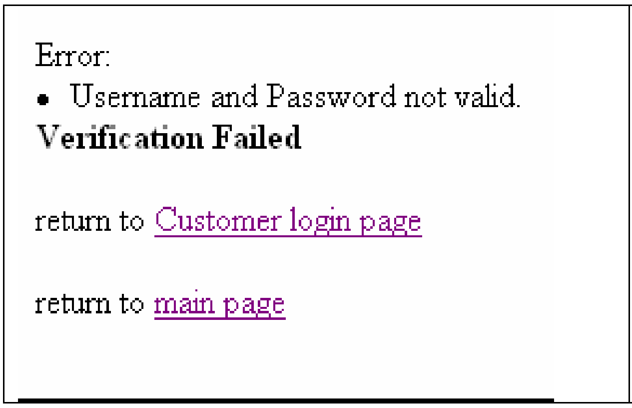


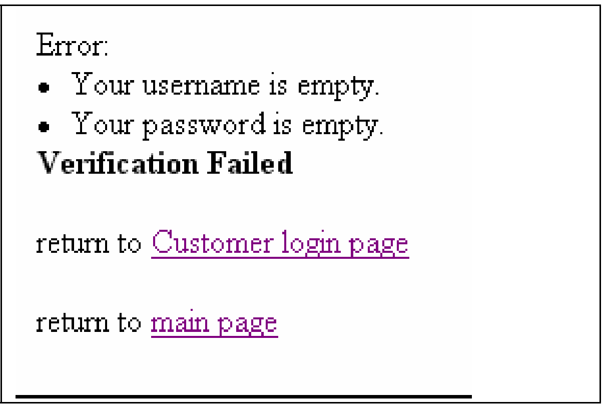
custlogin.php

Η ιστοσελίδα αυτή παρουσιάζει την φόρμα εισαγωγής στο σύστημα ενός πελάτη. Η φόρμα έχει τη μορφή που φαίνεται στο επόμενο σχήμα:



Ο χρήστης εδώ πρέπει να δώσει ένα όνομα χρήστη και έναν κωδικό πελάτη (τα οποία έχει αποκτήσει μετά από εγγραφή του) για να μπορέσει να δει τη σελίδα παραγγελιών των πελατών. Η φόρμα αποστέλλεται στο αρχείο **custverify.php**. Αν δοθεί κάτι λάθος το αρχείο αυτό σου βγάζει ένα λάθος και σου δίνει ένα σύνδεσμο για να επιστρέψεις στη φόρμα. Στα επόμενα δύο σχήματα έχουμε δύο περιπτώσεις όπου στην πρώτη δεν δόθηκε σωστό όνομα χρήστη-κωδικός και στη δεύτερη δεν συμπληρώθηκαν καθόλου τα πεδία. Σε όλες τις περιπτώσεις δίνεται πάντα και ένας σύνδεσμος για επιστροφή στην κυρίως σελίδα.



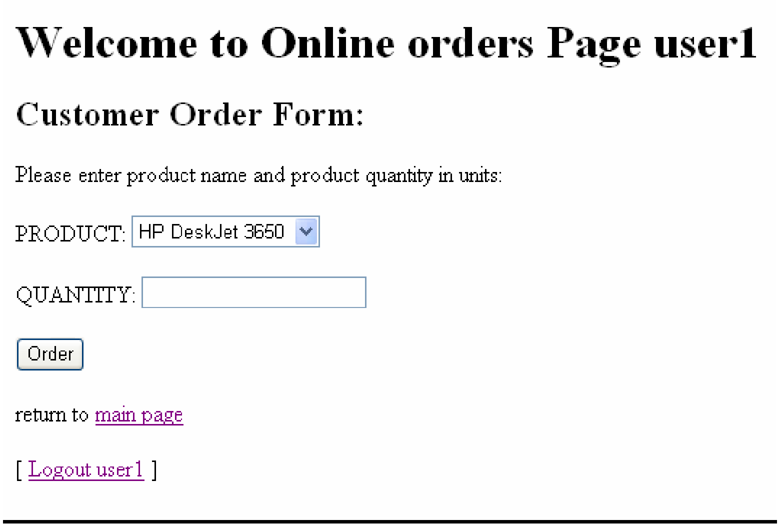


custverify.php

Η ιστοσελίδα αυτή ελέγχει την είσοδο στο σύστημα ενός πελάτη χρησιμοποιώντας τη συνάρτηση **custlogin_check()** από το αρχείο **functions.php**. Σε περίπτωση λάθους εμφανίζει ένα μήνυμα περιγραφής του λάθους και σου δίνει δύο συνδέσμους, έναν για να ξαναγυρίσεις στην φόρμα και να ξαναπροσπαθήσεις και έναν για να επιστρέψεις στην αρχική σελίδα (όπως φαίνεται στα δύο προηγούμενα σχήματα). Αν όλα πάνε καλά εμφανίζει την φόρμα παραγγελίας προϊόντων που είναι το αρχείο **custorder.php**.

custorder.php

Η ιστοσελίδα αυτή παρουσιάζει μία φόρμα για να εισάγει ένας πελάτης μία νέα παραγγελία όπως φαίνεται και στο επόμενο σχήμα. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **customersecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν πελάτης). Επίσης χρησιμοποιεί και την συνάρτηση **getproducts()** από το ίδιο αρχείο για να προβάλει σε μια λίστα όλα τα υπάρχοντα προϊόντα Ο πελάτης πρέπει να επιλέξει το όνομα του προϊόντος από τη λίστα και την ποσότητα του που θέλει να παραγγείλει. Η φόρμα αποστέλλεται στο αρχείο **verifycustorder.php**.



Επίσης υπάρχουν και οι επιλογές για να βγει από πελάτης (logout) και για να επιστρέψει στην αρχική σελίδα της εφαρμογής.

verifycustorder.php

Η ιστοσελίδα αυτή ελέγχει αν μπορεί να γίνει η παραγγελία του πελάτη. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **customersecure ()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν πελάτης). Επίσης χρησιμοποιεί και την συνάρτηση **customerorder_check()** από το ίδιο αρχείο για να κάνει τον έλεγχο και την εισαγωγή της παραγγελίας αν μπορεί να γίνει.

Στην περίπτωση που υπάρχει κάποιο λάθος προβάλει μια περιγραφή του και σου δίνει τις επιλογές να ξαναγυρίσεις στην φόρμα εισαγωγής παραγγελίας προϊόντος, να γυρίσεις στην αρχική σελίδα και την επιλογή να βγεις από πελάτης.

Δύο περιπτώσεις λάθους (δεν δόθηκε ποσότητα και η ποσότητα που δόθηκε δεν είναι διαθέσιμη) φαίνονται στα δύο επόμενα σχήματα.

- The quantity is empty.

Order Insertion Failed

The Order insertion Failed

return to insert order page

return to main page

[Logout user1]

- Product stock is not enough for your order (current stock=20).

Order Insertion Failed

The Order insertion Failed

return to insert order page

return to main page

[Logout user1]

Στο επόμενο σχήμα φαίνεται μία παραγγελία που ήταν επιτυχής.

The Order insertion was succesfull

Product: HP DeskJet 3650
Quantity: 5

return to insert order page

return to main page

[Logout user1]

custregform.htm

Η ιστοσελίδα αυτή παρουσιάζει την φόρμα εγγραφής στο σύστημα ενός νέου χρήστη, η οποία φαίνεται στο επόμενο σχήμα. Όποιος χρήστης θέλει να κάνει παραγγελίες πρέπει να εγγραφεί πρώτα μέσω αυτής της φόρμας. Αφού εγγραφεί το όνομα χρήστη και ο κωδικός που έδωσε θα είναι αυτοί που θα χρησιμοποιεί για να μπει στο σύστημα. Η φόρμα αποστέλλεται στο αρχείο **custsignin.php**.

Customer Sign-in Form

Please fill in all fields in order to sign-in:

Last Name:

First Name:

Address:

Username:

Password:
Confirm Password:

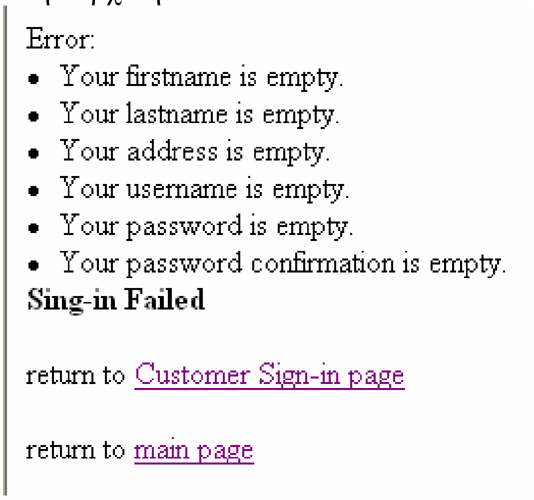
sign-in Reset

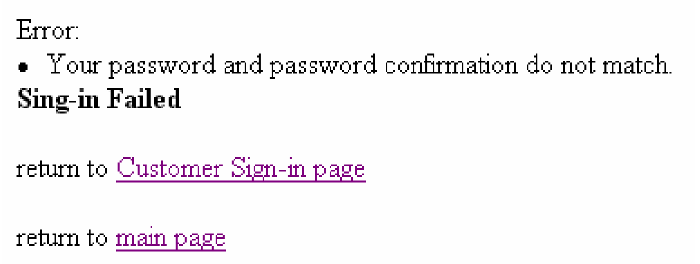
return to main page

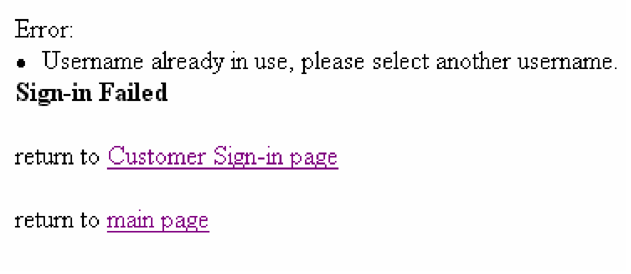
Επίσης η φόρμα δίνει και ένα σύνδεσμο για να επιστρέψεις στην αρχική σελίδα.

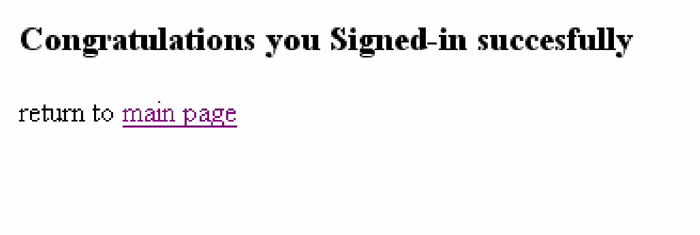
custsignin.php

Η ιστοσελίδα αυτή ελέγχει αν μπορεί να γίνει η εγγραφή ενός νέου πελάτη στο σύστημα. Χρησιμοποιεί τη συνάρτηση **signin_check()** από το αρχείο **functions.php** για να ελέγξει αν μπορεί να γίνει η εγγραφή. Σαν δεν μπορεί να γίνει η εισαγωγή εμφανίζεται περιγραφή του λάθους, αλλιώς γίνεται η εγγραφή και εμφανίζεται κατάλληλο μήνυμα επιτυχίας. Στα επόμενα σχήματα φαίνονται δύο περιπτώσεις που δεν έγινε η εισαγωγή και μία που πέτυχε.
Σε κάθε περίπτωση λάθους εμφανίζονται και δύο σύνδεσμοι για να ξαναπροσπαθήσεις και για να γυρίσεις στην αρχική σελίδα ενώ στην περίπτωση επιτυχίας εμφανίζεται ένας σύνδεσμος για να γυρίσεις στην αρχική σελίδα.









custlogout.php

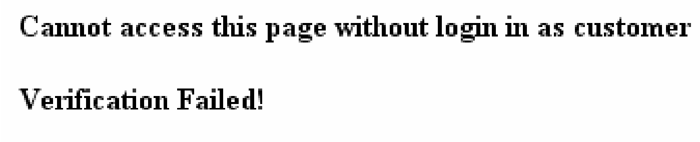
Η ιστοσελίδα αυτή υλοποιεί την έξοδο ενός πελάτη από την εφαρμογή. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **customersecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν πελάτης) και αν μπορεί καταστρέφει τη σύνοδο για το συγκεκριμένο χρήστη και εμφανίζει δύο συνδέσμους, τον πρώτο για να πάει ο χρήστης στην φόρμα εισαγωγής στην εφαρμογή σαν πελάτης και άλλη μία για να επιστρέψει στην αρχική σελίδα όπως φαίνεται στο επόμενο σχήμα.

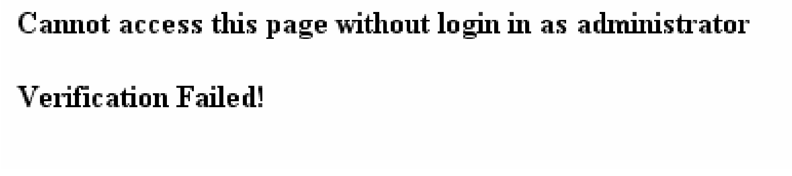
adminlogout.php

Η ιστοσελίδα αυτή υλοποιεί την έξοδο ενός διαχειριστή από την εφαρμογή. Η ιστοσελίδα αυτή χρησιμοποιεί τη συνάρτηση **adminsecure()** από το αρχείο **functions.php** για να ελέγξει αν ο χρήστης μπορεί να τη δει (αν έχει εισαχθεί στο σύστημα σαν διαχειριστής) και αν μπορεί καταστρέφει τη σύνοδο για το συγκεκριμένο διαχειριστή και εμφανίζει δύο συνδέσμους, τον πρώτο για να πάει ο χρήστης στην φόρμα εισαγωγής στην εφαρμογή σαν διαχειριστής και άλλη μία για να επιστρέψει στην αρχική σελίδα, όπως φαίνεται στο επόμενο σχήμα.

Ασφάλεια Προστατευμένων Σελίδων

Όλες οι ιστοσελίδες στις οποίες πρέπει να πάνε μόνο εξουσιοδοτημένοι χρήστες είναι προστατευμένες αφού ανάλογα την εξουσιοδότηση που χρειάζονται καλούν τη συνάρτηση customersecure() ή τη συνάρτηση adminsecure() από το αρχείο functions.php για να ελέγξουν αν ο χρήστης μπορεί να τη δει. Στην περίπτωση που δεν είναι εξουσιοδοτημένος τους εμφανίζεται ένα μήνυμα όπως φαίνεται στα επόμενα σχήματα για εξουσιοδότηση χρήστη και διαχειριστή αντίστοιχα.





Κώδικας (HTML – PHP) αρχείων Εφαρμογής

Ακολουθεί ο κώδικας των ιστοσελίδων/αρχείων.

functions.php

```
<?php
	$mysqluser = "root";
	$mysqlpassword = "";
	$mysqldb = "ordersdb";
	$serverhost = "localhost";
/*
 * This function checks if the new customer can sign-in.
 * First it checks if the customer did not filled all fields.
 * Next checks if the password and password confirmation are the same
 * Next checks if the username given already exist.
 * If anything of the above is true an error message is returned else the
 * new customer is inserted in table customers and "" is returned.
```

```
 */
function signin_check()
{
	global $mysqluser;
	global $mysqlpassword;
	global $mysqldb;
	global $serverhost;

	$paroxhstoixeiwn = true;

	$error = "";
	if ($_POST['firstname'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your firstname is empty.</li>";
	}
	else
		$firstname = $_POST['firstname'];

	if ($_POST['lastname'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your lastname is empty.</li>";
	}
	else
		$lastname = $_POST['lastname'];

	if ($_POST['address'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your address is empty.</li>";
	}
	else
		$address = $_POST['address'];

	if ($_POST['username'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your username is empty.</li>";
	}
	else
		$username = $_POST['username'];

	if ($_POST['password'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your password is empty.</li>";
	}
	else
		$password = $_POST['password'];

	if ($_POST['confirmpassword'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your password confirmation is empty.</li>";
	}
	else
		$confirmpassword = $_POST['confirmpassword'];

	if ($paroxhstoixeiwn)
```

```
        {
                if ($password != $confirmpassword)
                        $error .= "<li>Your password and password confirmation do not match.</li>";
        }

        if (trim($error)!="")
        {
                $error .= "<br><b>Sing-in Failed</b>";
                return $error;
        }
        else
        {
                /* Connecting to mysql server */
                $link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
                if (!$link)
                        die ('Cannot connect to server for verification : ' . mysql_error());

                /* selecting database from mysql server */
                $db_selected = mysql_select_db($mysqldb, $link);
                if (!$db_selected)
                die ('Cannot connect to database for verification : ' . mysql_error());

                /* selecting rows with the username and password given in login form */
                $sql = "SELECT * FROM customers where username='$username'";
                $result = mysql_query($sql);

                if (!$result)
                        die ('Cannot select from table for verification : ' . mysql_error());

                /* getting number of rows of the result query */
                $num = mysql_num_rows($result);

                if ($num >= 1)
                {
                        $error .= "<li>Username already in use, please select another username.</li>";
                        $error .= "<br><b>Sign-in Failed</b>";
                        return $error;
                }
                else
                {
                        $sql = "insert into customers (lastname, firstname, address, username, password) values
('$lastname', '$firstname', '$address', '$username', '$password')";
                        $result = mysql_query($sql);
                        if (!$result)
                                die ('Cannot insert in to table : ' . mysql_error());

                        /* getting number of rows added from the result query */
                        $num = mysql_affected_rows($link);

                        if ($num != 1)
                        {
                                die ('Cannot add user : ' . mysql_error());
                        }
                        else
                        {
                                return "";
                        }
                }
        }
}
```

```
/*
 * This function checks if a logged in customer order is valid and if yes it makes the order.
 * First it checks if the customer did not filled all fields.
 * Next checks if the quantity is a not a valid number.
 * Next checks if there is not enough stock of the product to do the order.
 * If anything of the above is true an error message is returned else the
 * customer order is inserted in table orders, the stock in table products is
 * updated and "" is returned.
 */
function customerorder_check()
{
	global $mysqluser;
	global $mysqlpassword;
	global $mysqldb;
	global $serverhost;

	$paroxhstoixeiwn = true;

	$error = "";
	if ($_POST['product'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>The product name is empty.</li>";
	}
	else
		$product = $_POST['product'];

	if ($_POST['quantity'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>The quantity is empty.</li>";
	}
	else
		$quantity = $_POST['quantity'];

	if ($paroxhstoixeiwn)
	{
		if (!ctype_digit($quantity))
		{
			$error .= "<li>The quantity is not a valid number.</li>";
		}
		else
		{
			if ($quantity <= 0)
				$error .= "<li>The quantity must be greater than 0</li>";
		}
	}

	if (trim($error)!="")
	{
		$error .= "<br><b>Order Insertion Failed</b>";
		return $error;
	}
	else
	{
		/* Connecting to mysql server */
		$link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
```

```
if (!$link)
        die ('Cannot connect to server for verification : ' . mysql_error());

/* selecting database from mysql server */
$db_selected = mysql_select_db($mysqldb, $link);
if (!$db_selected)
die ('Cannot connect to database for verification : ' . mysql_error());

/* selecting rows with the username and password given in login form */
$sql = "SELECT * FROM products where product='$product'";
$result = mysql_query($sql);

if (!$result)
        die ('Cannot select from table for verification : ' . mysql_error());

/* getting number of rows of the result query */
$num = mysql_num_rows($result);

if ($num != 1)
{
        $error .= "<li>Could not select product from table.</li>";
        $error .= "<br><b>Order Insertion Failed</b>";
        return $error;
}


if ($r = mysql_fetch_row($result))
{
        $currentstock = $r[2];
        $productid = $r[0];
}
else
{
        $error .= "<li>Could not read products current stock.</li>";
        $error .= "<br><b>Order Insertion Failed</b>";
        return $error;
}

/* if the stock is not enough return error */
if ($currentstock < $quantity)
{
        /* close connection */
        mysql_close($link);
        $error .= "<li>Product stock is not enough for your order (current
stock=$currentstock).</li>";
        $error .= "<br><b>Order Insertion Failed</b>";
        return $error;
}

$customerid = $_SESSION['customerid'];
$sql = "insert into orders (customer_id, product_id, quantity) values ($customerid, $productid,
$quantity)";
$result = mysql_query($sql);
if (!$result)
        die ('Cannot insert in to table : ' . mysql_error());
/* getting number of rows added from the result query */
$num = mysql_affected_rows($link);

if ($num != 1)
{
```

```
				die ('Cannot insert product : ' . mysql_error());
			}
			else
			{
				$currentstock = $currentstock - $quantity;
				$sql = "Update products set stock=$currentstock where product='$product'";
				$result = mysql_query($sql);
				if (!$result)
					die ('Cannot update table : ' . mysql_error());

				/* getting number of rows added from the result query */
				$num = mysql_affected_rows($link);

				if ($num != 1)
				{
					die ('Cannot update product $num: ' . mysql_error());
				}
				else
				{
					/* close connection */
					mysql_close($link);
					return "";
				}
			}
		}
}

/*
 * This function checks if a logged in administrator can insert a new product.
 * First it checks if the administrator did not filled all fields.
 * Next checks if the quantity is a not a valid number.
 * Next checks if the product already exists.
 * If anything of the above is true an error message is returned else the
 * new product is inserted in table products and "" is returned.
 */
function insertproduct_check()
{
		global $mysqluser;
		global $mysqlpassword;
		global $mysqldb;
		global $serverhost;

		$paroxhstoixeiwn = true;

		$error = "";
		if ($_POST['product'] == null)
		{
			$paroxhstoixeiwn = false;
			$error .= "<li>The product name is empty.</li>";
		}
		else
			$product = $_POST['product'];

		if ($_POST['quantity'] == null)
		{
			$paroxhstoixeiwn = false;
			$error .= "<li>The quantity is empty.</li>";
		}
		else
			$quantity = $_POST['quantity'];
```

```
	if ($paroxhstoixeiwn)
	{
		if (!ctype_digit($quantity))
		{
			$error .= "<li>The quantity is not a valid number.</li>";
		}
		else
		{
			if ($quantity <= 0)
				$error .= "<li>The quantity must be greater than 0</li>";
		}
	}

	if (trim($error)!="")
	{
		$error .= "<br><b>Product Insertion Failed</b>";
		return $error;
	}
	else
	{
		/* Connecting to mysql server */
		$link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
		if (!$link)
			die ('Cannot connect to server for verification : ' . mysql_error());

		/* selecting database from mysql server */
		$db_selected = mysql_select_db($mysqldb, $link);
		if (!$db_selected)
		die ('Cannot connect to database for verification : ' . mysql_error());

		/* selecting rows with the username and password given in login form */
		$sql = "SELECT * FROM products where product='$product'";
		$result = mysql_query($sql);

		if (!$result)
			die ('Cannot select from table for verification : ' . mysql_error());

		/* getting number of rows of the result query */
		$num = mysql_num_rows($result);

		if ($num >= 1)
		{
			/* close connection */
			mysql_close($link);
			$error .= "<li>The product already exist, please use Product Update Form to update stock of
existing products.</li>";
			$error .= "<br><b>Product Insertion Failed</b>";
			return $error;
		}
		else
		{
			$sql = "insert into products (product, stock) values ('$product', $quantity)";
			$result = mysql_query($sql);
			if (!$result)
				die ('Cannot insert in to table : ' . mysql_error());

			/* getting number of rows added from the result query */
			$num = mysql_affected_rows($link);
```

```
					if ($num != 1)
					{
						die ('Cannot insert product : ' . mysql_error());
					}
					else
					{
						/* close connection */
						mysql_close($link);
						return "";
					}
			}
		}
}


/*
 * This function checks if a logged in administrator can update an existing product.
 * First it checks if the administrator did not filled all fields.
 * Next checks if the quantity is a not a valid number.
 * Next checks if the product does not exists.
 * If anything of the above is true an error message is returned else the
 * product is updated in table products by adding the quantity to the stock and "" is returned.
 */
function updateproductstock_check()
{
		global $mysqluser;
		global $mysqlpassword;
		global $mysqldb;
		global $serverhost;

		$paroxhstoixeiwn = true;

		$error = "";
		if ($_POST['product'] == null)
		{
			$paroxhstoixeiwn = false;
			$error .= "<li>The product name is empty.</li>";
		}
		else
			$product = $_POST['product'];

		if ($_POST['quantity'] == null)
		{
			$paroxhstoixeiwn = false;
			$error .= "<li>The quantity is empty.</li>";
		}
		else
			$quantity = $_POST['quantity'];

		if ($paroxhstoixeiwn)
		{
			if (!ctype_digit($quantity))
			{
				$error .= "<li>The quantity is not a valid number.</li>";
			}
			else
			{
				if ($quantity <= 0)
					$error .= "<li>The quantity must be greater than 0</li>";
```

```
			}
		}

		if (trim($error)!="")
		{
			$error .= "<br><b>Product Update Failed</b>";
			return $error;
		}
		else
		{
			/* Connecting to mysql server */
			$link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
			if (!$link)
				die ('Cannot connect to server for verification : ' . mysql_error());

			/* selecting database from mysql server */
			$db_selected = mysql_select_db($mysqldb, $link);
			if (!$db_selected)
			die ('Cannot connect to database for verification : ' . mysql_error());

			/* selecting rows with the username and password given in login form */
			$sql = "SELECT * FROM products where product='$product'";
			$result = mysql_query($sql);

			if (!$result)
				die ('Cannot select from table for verification : ' . mysql_error());

			/* getting number of rows of the result query */
			$num = mysql_num_rows($result);

			if ($num != 1)
			{
				/* close connection */
				mysql_close($link);
				$error .= "<li>The product does not exist, please use Product Insertion Form to insert new
products.</li>";
				$error .= "<br><b>Product Update Failed</b>";
				return $error;
			}
			else
			{
				if ($r = mysql_fetch_row($result))
				{
					$currentstock = $r[2];
				}
				else
				{
					$error .= "<li>Could not read products current stock.</li>";
					$error .= "<br><b>Product Update Failed</b>";
					return $error;
				}

				$currentstock += $quantity;
				$sql = "Update products set stock=$currentstock where product='$product'";
				$result = mysql_query($sql);
				if (!$result)
					die ('Cannot update table : ' . mysql_error());

				/* getting number of rows added from the result query */
				$num = mysql_affected_rows($link);
```

```
				if ($num != 1)
				{
					die ('Cannot update product $num: ' . mysql_error());
				}
				else
				{
					/* close connection */
					mysql_close($link);
					return "";
				}
			}
		}
}

/*
 * This function checks if a customer is authorized to view a page (i.e. if is logged in).
 * This is done if the variables that were registered and set to some value at login time have
 * values. If is not authorized an error message is printed and the processing of the page is stopped.
 * session_start() must be call before calling this function
 */
function customersecure()
{
	if ( $_SESSION['user'] == "" || $_SESSION['loggedIn'] != "true" )
	{
		die('<b>Cannot access this page without login in as customer</b><p><b>Verification Failed!</b>');
	}
}


/*
 * This function checks if an administrator is authorized to view a page (i.e. if is logged in).
 * This is done by checking if the variables that were registered and set to some value at login time have
 * values. If is not authorized an error message is printed and the processing of the page is stopped.
 * session_start() must be call before calling this function
 */
function adminsecure()
{
	if ( $_SESSION['admin'] == "" || $_SESSION['loggedIn'] != "true" )
	{
		die('<b>Cannot access this page without login in as administrator</b><p><b>Verification 
Failed!</b>');
	}
}


/*
 * This function checks if a customer can log in.
 * First it checks if the customer did not filled all fields.
 * Next checks if the pair username-password given is not valid.
 * If anything of the above is true an error message is returned else the
 * customer is logged in by starting a session and registering and giving values
 * to three variabes user, customer_id and loggedIn and "" is returned.
 */
function customerlogin_check ()
{
	global $mysqluser;
	global $mysqlpassword;
	global $mysqldb;
	global $serverhost;
```

```
$paroxhstoixeiwn = true;

/* Check if user filled in form fields */
$error = "";
if ($_POST['username'] == null)
{
        $paroxhstoixeiwn = false;
        $error .= "<li>Your username is empty.</li>";
}
else
        $username = $_POST['username'];

if ($_POST['password'] == null)
{
        $paroxhstoixeiwn = false;
        if (trim($password) == "") $error .= "<li>Your password is empty.</li>";
}
else
        $password = $_POST['password'];

if (trim($error)!="")
{
        $error .= "<br><b>Verification Failed</b>";
        return $error;
}

/* Connecting to mysql server */
$link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
if (!$link)
        die ('Cannot connect to server for verification : ' . mysql_error());

/* selecting database from mysql server */
$db_selected = mysql_select_db($mysqldb, $link);
if (!$db_selected)
   die ('Cannot connect to database for verification : ' . mysql_error());

/* selecting rows with the username and password given in login form */
$sql = "SELECT * FROM customers where username='$username' and password='$password'";
$result = mysql_query($sql);

if (!$result)
        die ('Cannot select from table for verification : ' . mysql_error());

/* getting number of rows of the result query */
$num = mysql_num_rows($result);

/* check if one row exists that matches customer */
if ($num == 1)
{
        $r = mysql_fetch_row($result);

        session_start();
        session_register("user");
        session_register("loggedIn");
        $_SESSION['user'] = $_POST['username'];
        $_SESSION['customerid'] = $r[0];
        $_SESSION['loggedIn'] = "true";

        /* close connection */
```

```
		mysql_close($link);

		return "";
	}
	else
	{
		/* close connection */
		mysql_close($link);

		$error .= "<li>Username and Password not valid.</li>";
		$error .= "<br><b>Verification Failed</b>";
		return $error;
	}
}

/*
 * This function checks if an administrator can log in.
 * First it checks if the administrator did not filled all fields.
 * Next checks if the pair username-password given is not valid.
 * If anything of the above is true an error message is returned else the
 * administrator is logged in by starting a session and registering and giving
 * values to two variabes admin and loggedIn and "" is returned.
 */
function adminlogin_check ()
{
	global $mysqluser;
	global $mysqlpassword;
	global $mysqldb;
	global $serverhost;

	$paroxhstoixeiwn = true;

	/* Check if user filled in form fields */
	$error = "";
	if ($_POST['username'] == null)
	{
		$paroxhstoixeiwn = false;
		$error .= "<li>Your username is empty.</li>";
	}
	else
		$username = $_POST['username'];

	if ($_POST['password'] == null)
	{
		$paroxhstoixeiwn = false;
		if (trim($password) == "") $error .= "<li>Your password is empty.</li>";
	}
	else
		$password = $_POST['password'];

	if (trim($error)!="")
	{
		$error .= "<br><b>Verification Failed</b>";
		return $error;
	}

	/* Connecting to mysql server */
	$link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
	if (!$link)
		die ('Cannot connect to server for verification : ' . mysql_error());
```

```
        /* selecting database from mysql server */
        $db_selected = mysql_select_db($mysqldb, $link);
        if (!$db_selected)
           die ('Cannot connect to database for verification : ' . mysql_error());

        /* selecting rows with the username and password given in login form */
        $sql = "SELECT * FROM administrators where username='$username' and password='$password'";
        $result = mysql_query($sql);

        if (!$result)
                die ('Cannot select from table for verification : ' . mysql_error());

        /* getting number of rows of the result query */
        $num = mysql_num_rows($result);

        /* check if one row exists that matches administrator */
        if ($num == 1)
        {
                session_start();
                session_register("admin");
                session_register("loggedIn");
                $_SESSION['admin'] = $_POST['username'];
                $_SESSION['loggedIn'] = "true";

                /* close connection */
                mysql_close($link);

                return "";
        }
        else
        {
                /* close connection */
                mysql_close($link);

                $error .= "<li>Username and Password not valid.</li>";
                $error .= "<br><b>Verification Failed</b>";
                return $error;
        }
}


/*
 * This function returns the result rowset of an sql query which selects all available
 * products in table products.
 * It is used to get the products and display them in a list in the insert order form of a logged in
 * customer and in a list in the update product form of a logged in administrator.
 *
 */
function getproducts()
{
        global $mysqluser;
        global $mysqlpassword;
        global $mysqldb;
        global $serverhost;

        /* Connecting to mysql server */
        $link = mysql_connect($serverhost, $mysqluser, $mysqlpassword);
        if (!$link)
                die ('Cannot connect to server for verification : ' . mysql_error());
```

```
        /* selecting database from mysql server */
        $db_selected = mysql_select_db($mysqldb, $link);
        if (!$db_selected)
           die ('Cannot connect to database for verification : ' . mysql_error());

        /* selecting rows with the username and password given in login form */
        $sql = "SELECT * FROM products order by product";
        $result = mysql_query($sql);

        if (!$result)
                die ('Cannot select from table : ' . mysql_error());

        return $result;
}

?>
```

index.htm

```
<html>
<head>
<title>ON-LINE ORDERS - MAIN PAGE</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<h1>Welcome to Online orders Page</h1>
<p>
Please select one of the three choices:
<p>
<ul>
  <li> <a href="adminlogin.php">Administrators login page</a> (click here to insert new products or update product stocks)</li>
  <li><a href="custlogin.php">Customers login page </a> (make a new order if you have already signed in)</li>
  <li><a href="custregform.htm">Customer sign-in page</a> (sign-in first before ordering products)</li>
</ul>
</body>
</html>
```

adminlogin.php

```
<HTML>
<head>
<title>ON-LINE ORDERS - ADMINISTRATOR LOGIN PAGE</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<BODY>

<h2>Administrator Login Form:</h2>
<p>
Please enter your username and password:
<p>
<FORM action="adminverify.php" METHOD="post">
USERNAME: <INPUT TYPE="text" maxlength=20 NAME="username">
<p>
PASSWORD: <INPUT TYPE="password" maxlength=20 NAME="password">
<BR>
<p>
<INPUT TYPE="submit" VALUE="Login">
```

```
</FORM>
<p>
<p>
return to <a href="index.htm">main page</a>
</BODY>
</HTML>
```

adminverify.php

```
<?php

        include ("functions.php");
        $url = "adminindex.php";

        $error = adminlogin_check();
        if (trim($error)=="")
        {
                header("Location: $url");
        }
        else
        {
                echo "<HTML>";
                echo "<head>";
                echo "<title>ON-LINE ORDERS - ADMINISTRATOR LOGIN PAGE</title>";
                echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
                echo "</head>";
                echo "<BODY>";
                echo "Error:$error";
                echo "<p>";
                echo "<p>";
                echo "<p>";
                echo "return to <a href='adminlogin.php'>Administrator login page</a>";
                echo "<p>";
                echo "return to <a href='index.htm'>main page</a>";
                echo "</BODY>";
                echo "</HTML>";
        }
?>
```

adminindex.php

```
<?php
include("functions.php");
session_start();
adminsecure();

echo "<html>";
echo "<head>";
echo "<title>ON-LINE ORDERS - ADMIN MAIN PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<body>";
echo "<h1>Welcome to Online orders Page " . $_SESSION['admin'] . "</h1>";
echo "<p>";
echo "Please select one of the two choices:";
echo "<p>";
echo "<ul>";
echo "<li><a href='insertproducts.php'>Insert new Product</a> (click here to insert new products)</li>";
```

```
echo "<li><a href='updateproducts.php'>Update existing Product stocks </a> (click here to update product stocks)</li>";
echo "</ul>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "[ <a href='adminlogout.php'>Logout " . $_SESSION['admin'] . "</a> ]";
echo "</body>";
echo "</html>";

?>
```

insertproducts.php

```
<?php
include("functions.php");
session_start();
adminsecure();

echo "<HTML>";
echo "<head>";
echo "<title>ON-LINE ORDERS - ADMINISTRATOR INSERT PRODUCT PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<BODY>";
echo "<p>";
echo "<h1>Welcome to insert product Page " . $_SESSION['admin'] . "</h1>";
echo "<p>";
echo "<h2>Admin Insert Product Form:</h2>";
echo "<p>";
echo "Please enter product name and product quantity in units:";
echo "<p>";
echo "<FORM action='verifyproductinsertion.php' METHOD='post'>";
echo "PRODUCT: <INPUT TYPE='text' MAXLENGTH=50 NAME='product'>";
echo "</SELECT>";
echo "<p>";
echo "QUANTITY: <INPUT TYPE='text' MAXLENGTH=5 NAME='quantity'>";
echo "<BR>";
echo "<p>";
echo "<INPUT TYPE='submit' VALUE='Insert'>";
echo "</FORM>";
echo "<p>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "return to <a href='adminindex.php'>administrator main page</a>";
echo "<p>";
echo "[ <a href='adminlogout.php'>Logout " . $_SESSION['admin'] . "</a> ]";
echo "</BODY>";
echo "</HTML>";
?>
```

updateproducts.php

```
<?php
include("functions.php");
session_start();
adminsecure();

echo "<HTML>";
echo "<head>";
```

```
echo "<title>ON-LINE ORDERS - ADMINISTRATOR UPDATE PRODUCT PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<BODY>";
echo "<p>";
echo "<h1>Welcome to update product Page ";
echo $_SESSION['admin'];
echo "</h1>";
echo "<p>";
echo "<h2>Admin Update Product Form:</h2>";
echo "<p>";
echo "Please select product name and product quantity in units to add to product stock:";
echo "<p>";
echo "<FORM action='verifyproductupdate.php' METHOD='post'>";
echo "PRODUCT TO UPDATE: <SELECT NAME='product'>";

$result = getproducts();
$num_results = mysql_num_rows($result);

for ($i=0; $i < $num_results; ++$i)
{
	$r = mysql_fetch_array($result);
	echo "<OPTION VALUE=\"" . $r['product'] . "\">" . $r['product'] . "</option>";
}

echo "</SELECT>";
echo "<p>";
echo "QUANTITY TO ADD: <INPUT TYPE='text' MAXLENGTH=5 NAME='quantity'>";
echo "<BR>";
echo "<p>";
echo "<INPUT TYPE='submit' VALUE='Update'>";
echo "</FORM>";
echo "<p>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "return to <a href='adminindex.php'>administrator main page</a>";
echo "<p>";
echo "[ <a href='adminlogout.php'>Logout " . $_SESSION['admin'] . "</a> ]";
echo "</BODY>";
echo "</HTML>";
?>
```

verifyproductinsertion.php

```
<?php
include("functions.php");
session_start();
adminsecure();

echo "<html>";
echo "<head>";
echo "<title>ON-LINE ORDERS - ADMIN INSERT PRODUCT PAGE VERIFICATION</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<body>";

$error = insertproduct_check();
if (trim($error)=="")
{
	echo "<h1>The insertion was succesfull</h1>";
```

```
	echo "<b>Product: " . $_POST['product'] . "<b><br><b>Quantity: " . $_POST['quantity'] . "<b>";
}
else
{
	echo $error;
	echo "<h1>The insertion Failed</h1>";
}
echo "<p>";
echo "return to <a href='insertproducts.php'>insert product page</a>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "[ <a href='adminlogout.php'>Logout {$_SESSION['admin']}</a> ]";
echo "</BODY>";
echo "</HTML>";
?>
```

verifyproductupdate.php

```
<?php
include("functions.php");
session_start();
adminsecure();

echo "<html>";
echo "<head>";
echo "<title>ON-LINE ORDERS - ADMIN UPDATE PRODUCT STOCK PAGE VERIFICATION</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<body>";

$error = updateproductstock_check();
if (trim($error)=="")
{
	echo "<h1>The update was succesfull</h1>";
	echo "<b>Product: " . $_POST['product'] . "<b><br><b>Quantity Added: " . $_POST['quantity'] . "<b>";
}
else
{
	echo $error;
	echo "<h1>The updateFailed</h1>";
}
echo "<p>";
echo "return to <a href='updateproducts.php'>update product page</a>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "[ <a href='adminlogout.php'>Logout " . $_SESSION['admin'] . "</a> ]";
echo "</BODY>";
echo "</HTML>";
?>
```

custlogin.php

```
<HTML>
<BODY>
<h2>Customer Login Form:</h2>
<p>
Please enter your username and password:
<p>
<FORM action="custverify.php" METHOD="post">
```

```
USERNAME: <INPUT TYPE="text" maxlength=20 NAME="username">
<p>
PASSWORD: <INPUT TYPE="password" maxlength=20 NAME="password">
<BR>
<p>
<INPUT TYPE="submit" VALUE="Login">
</FORM>
<p>
<p>
return to <a href="index.htm">main page</a>
</BODY>
</HTML>
```

custverify.php

```
<?php

        include ("functions.php");
        $url = "custorder.php";

        $error = customerlogin_check();
        if (trim($error)=="")
        {
                header("Location: $url");
        }
        else
        {
                echo "<HTML>";
                echo "<BODY>";
                echo "Error:$error";
                echo "<p>";
                echo "<p>";
                echo "return to <a href='custlogin.php'>Customer login page</a>";
                echo "<p>";
                echo "return to <a href='index.htm'>main page</a>";
                echo "</BODY>";
                echo "</HTML>";
        }
?>
```

custorder.php

```
<?php
include("functions.php");
session_start();
customersecure();

echo "<HTML>";
echo "<head>";
echo "<title>ON-LINE ORDERS - CUSTOMER ORDER PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<BODY>";
echo "<p>";
echo "<h1>Welcome to Online orders Page " . $_SESSION['user'] . "</h1>";
echo "<p>";
echo "<h2>Customer Order Form:</h2>";
echo "<p>";
echo "Please enter product name and product quantity in units:";
echo "<p>";
echo "<FORM action='verifycustorder.php' METHOD='post'>";
```

```
echo "PRODUCT: <SELECT NAME='product'>";

$result = getproducts();
$num_results = mysql_num_rows($result);

for ($i=0; $i < $num_results; ++$i)
{
        $r = mysql_fetch_array($result);
        echo "<OPTION VALUE=\"" . $r['product'] . "\">" . $r['product'] . "</option>";
}
echo "</SELECT>";
echo "<p>";
echo "QUANTITY: <INPUT TYPE='text' MAXLENGTH=5 NAME='quantity'>";
echo "<BR>";
echo "<p>";
echo "<INPUT TYPE='submit' VALUE='Order'>";
echo "</FORM>";
echo "<p>";
echo "<p>";
echo "return to <a href='index.htm'>main page</a>";
echo "<p>";
echo "[ <a href='custlogout.php'>Logout " . $_SESSION['user'] . "</a> ]";
echo "</BODY>";
echo "</HTML>";
?>
```

verifycustorder.php

```
<?php
        include("functions.php");
        session_start();
        customersecure();

        echo "<html>";
        echo "<head>";
        echo "<title>ON-LINE ORDERS - CUSTOMER INSERT ORDER PAGE VERIFICATION</title>";
        echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
        echo "</head>";
        echo "<body>";

        $error = customerorder_check();
        if (trim($error)=="")
        {
                echo "<h1>The Order insertion was succesfull</h1>";
                echo "<b>Product: " . $_POST['product'] ."<b><br><b>Quantity: " .$_POST['quantity'] ."<b>";
        }
        else
        {
                echo $error;
                echo "<h1>The Order insertion Failed</h1>";

        }
        echo "<p>";
        echo "return to <a href='custorder.php'>insert order page</a>";
        echo "<p>";
        echo "return to <a href='index.htm'>main page</a>";
        echo "<p>";
        echo "[ <a href='custlogout.php'>Logout {$_SESSION['user']}</a> ]";
        echo "</BODY>";
        echo "</HTML>";
?>
```

custregform.htm

```
<HTML>
<head>
<title>ON-LINE ORDERS - CUSTOMER SIGN-IN PAGE</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<BODY>

<h3>Customer Sign-in Form</h3>
<p>
Please fill in all fields in order to sign-in:
<p>
<FORM action="custsignin.php" METHOD="post">
<p>
Last Name: <INPUT TYPE="text" MAXLENGTH=30 NAME="lastname">
<p>
First Name: <INPUT TYPE="text" MAXLENGTH=30 NAME="firstname">
<p>
Address: <INPUT TYPE="text" MAXLENGTH=50 NAME="address">
<p>
Username: <INPUT TYPE="text" MAXLENGTH=20 NAME="username">
<p>
Password: <INPUT TYPE="password" MAXLENGTH=20 NAME="password">
<br>
Confirm Password: <INPUT TYPE="password" MAXLENGTH=20 NAME="confirmpassword">
<p>
<INPUT TYPE="submit" VALUE="sign-in">
<INPUT TYPE="reset" name="reset" value="Reset">
</FORM>
<p>
<p>
return to <a href="index.htm">main page</a>

</BODY>
</HTML>
```

custsignin.php

```
<?php
	include ("functions.php");

	$error = signin_check();
	if (trim($error)=="")
	{
		echo "<HTML>";
		echo "<BODY>";
		echo "<p>";
		echo "<h3>Congratulations you Signed-in succesfully</h3>";
		echo "<p>";
		echo "return to <a href='index.htm'>main page</a>";
		echo "</BODY>";
		echo "</HTML>";
	}
	else
	{
		echo "<HTML>";
		echo "<BODY>";
		echo "Error:$error";
		echo "<p>";
		echo "<p>";
```

```
                    echo "return to <a href='custregform.htm'>Customer Sign-in page</a>";
                    echo "<p>";
                    echo "return to <a href='index.htm'>main page</a>";
                    echo "</BODY>";
                    echo "</HTML>";
          }
?>
```

custlogout.php

```
<?php
include ("functions.php");
session_start();
customersecure();
$user = $_SESSION['user'];

//destroy session so that user cannot enter protected pages
session_unset();
session_destroy();

echo "<html>";
echo "<head>";
echo "<title>ON-LINE ORDERS - CUSTOMER LOGOUT PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<body>";
echo "<h1>You have successfully logged out " . $user . "</h1>";
echo "<p>";
echo "Click <a href='custlogin.php'>here</a> to log back in.";
echo "<p>";
echo "Return to <a href='index.htm'>main page</a>";
echo "</body>";
echo "</html>";
?>
```

adminlogout.php

```
<?php
include ("functions.php");
session_start();
adminsecure();
$admin = $_SESSION['admin'];

//destroy session so that user cannot enter protected pages
session_unset();
session_destroy();

echo "<html>";
echo "<head>";
echo "<title>ON-LINE ORDERS - ADMINISTRATOR LOGOUT PAGE</title>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>";
echo "</head>";
echo "<body>";
echo "<h1>You have successfully logged out $admin </h1>";
echo "<p>";
echo "Click <a href='adminlogin.php'>here</a> to log back in.";
echo "<p>";
echo "Return to <a href='index.htm'>main page</a>";
echo "</body>";
echo "</html>";
?>
```